What is claimed is:

1. In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method for booting the operating system comprising:

computing a cryptographic function of at least a portion of the operating system; and

setting the software identity register to a result of the computed cryptographic function.

2. The method as recited in claim 1, further comprising defining a secure storage space, access to which is based in part on the result set in the software identity register.

3. In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method for booting the operating system comprising:

executing an atomic operation to set an identity of the operating system into the software identity register of the CPU, wherein in an event that the atomic operation completes correctly, the software identity register contains the identity of the operating system and in an event that the atomic operation fails to complete correctly, the software identity register contains a value other than the identity of the operating system; and

examining a content of the software identity register to verify the identity of the operating system.

4. The method as recited in claim 3, wherein the identity comprises a public key of a correctly signed block of code from the operating system, and examining a content of the software identity register comprises verifying a signature of the signed block of code against the public key.

5. The method as recited in claim 3, wherein the identity comprises a hash digest of a block of code from the operating system, and examining a content of the software identity register comprises hashing the block of code.

6. The method as recited in claim 3, further comprising appending at least a portion of the identity to a boot log.

7. The method as recited in claim 3, further comprising authenticating additional blocks of code.

8. The method as recited in claim 3, further comprising:

appending at least a portion of the identity to a boot log;

authenticating additional blocks of code; and

appending identities of the additional blocks of code to the boot log.

9. The method as recited in claim 3, further comprising generating a storage key for encrypting data to be stored on the computer system from a seed based in part on the identity of the operating system.

10. The method as recited in claim 3, further comprising encrypting data using the storage key and storing the encrypted data on the computer system.

11. In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a method comprising:

identifying a boot block of code in the OS that uniquely describes the OS;

creating an identity of the OS from the boot block; and

executing an atomic operation to set the identity of the operating system into the software identity register of the CPU, wherein in an event that the atomic operation completes correctly, the software identity register contains the identity of the operating system.

12. The method as recited in claim 11, wherein creating an identity of the OS comprises signing the boot block using a private key from a key pair to form a signature, the signature and a corresponding public key from the key pair forming the OS identity.

13. The method as recited in claim 11, wherein creating an identity of the OS comprises hashing the boot block to form a digest, the digest forming the OS identity.

14. The method as recited in claim 11, further comprising appending at least a portion of the identity to a boot log.

15. The method as recited in claim 11, further comprising authenticating additional blocks of code.

16. The method as recited in claim 11, further comprising:

appending at least a portion of the identity to a boot log;

authenticating additional blocks of code; and

appending identities of the additional blocks of code to the boot log.

17. The method as recited in claim 11, further comprising generating a storage key for encrypting data to be stored on the computer system from a seed based in part on the identity of the OS.

18. The method as recited in claim 17, further comprising encrypting data using the storage key and storing the encrypted data on the computer system.

19. In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys and a software identity register that holds an identity of the operating system, a method comprising:

creating an identity of the OS containing the identity from the software identity register, information describing the operating system, and the CPU public key; and

signing the OS certificate using the CPU private key.

20. The method as recited in claim 19, further comprising submitting the signed OS certificate over a network to a third party to prove an identity of the operating system to the third party.

21. The method as recited in claim 19, wherein creating an identity of the OS comprises forming the OS certificate with one or more items from a boot log containing identities of software components that are executing on the CPU.

22. A method for establishing a chain of trust between a subscriber unit and a content provider, the subscriber unit having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys, a manufacturer certificate supplied by a manufacturer of the CPU, and a software identity register that holds an identity of the operating system, the method comprising:

submitting a request from the subscriber unit to the content provider, the request specifying a particular content;

generating, at the content provider, a challenge nonce;

returning the challenge nonce from the content provider to the subscriber unit;

forming, at the subscriber unit, an OS certificate containing the identity from the software identity register, information describing the operating system, the challenge nonce, and the CPU public key and signing the OS certificate using the CPU private key;

passing the OS certificate and the CPU manufacturer certificate from the subscriber unit to the content provider; and

evaluating, at the content provider, the OS certificate and the CPU manufacturer at the content provider to determine whether to reject or fulfill the request.

23. The method as recited in claim 22, wherein forming an OS certificate comprises forming the OS certificate with one or more items from a boot log containing identities of software components that are executing on the CPU.

24. The method as recited in claim 22, wherein evaluating the OS certificate comprises determining whether to trust the identity in the OS certificate.

25. The method as recited in claim 22, wherein evaluating the OS certificate comprises determining whether the challenge nonce returned in the OS certificate is the challenge nonce generated by the content provider.

26. The method as recited in claim 22, wherein evaluating the OS certificate comprises verifying the signature on the OS certificate using the CPU public key contained in the OS certificate.

27. The method as recited in claim 22, wherein evaluating the OS certificate comprises determining whether the OS certificate and the manufacturer certificate contain an identical CPU public key.

28. The method as recited in claim 22, wherein evaluating the OS certificate comprises verifying a manufacturer signature on the manufacturer certificate.

29. The method as recited in claim 22, wherein evaluating the OS certificate comprises determining whether to trust the manufacture of the CPU.

30. The method as recited in claim 22, further comprising downloading the content specified in the request in an event that the content provider elects to fulfill the request.

31. The method as recited in claim 30, further comprising encrypting the content using a storage key derived in part from the identity of the operating system.

32. In a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys and a software identity register that holds an identity of the operating system, the computer system further maintaining a boot log that holds identities of software components that are currently executing, a method comprising:

forming a generator seed from a CPU-specific secret, a user-supplied seed, and OS-specific data from the boot log; and

generating a storage key based on a function of the generator seed.

33. The method as recited in claim 32, wherein forming a generator key and generating a storage key comprises creating a storage key SK as follows:

SK=SHA(CPU-specific secret, OS-specific data, seed).

34. The method as recited in claim 32, further comprising encrypting data using the storage key.

35. A computer comprising:

a memory;

a central processing unit (CPU) coupled to the memory, the CPU having a software identity register;

an operating system stored in the memory, the operating system having a block of code; and

the operating system being booted for execution on the CPU according to a sequence that begins with an atomic operation, wherein in an event that the atomic operation completes correctly, the software identity register is set to the identity of the operating system.

36. The computer as recited in claim 35, wherein the identity comprises a digital signature on a block of code from the operating system.

37. The computer as recited in claim 35, wherein the identity comprises a hash digest of a block of code from the operating system.

38. The computer as recited in claim 35, wherein the CPU holds a manufacturer certificate signed by a manufacturer of the CPU.

39. The computer as recited in claim 35, further comprising a boot log, wherein the CPU appends the identity of the operating system to the boot log in the event that the atomic operation completes correctly.

40. The computer as recited in claim 35, wherein the CPU is assigned a pair of public and private keys, and CPU is configured to create an OS certificate containing the identity in the software identity register, information describing the operating system, and the CPU public key, the CPU signing the OS certificate using the CPU private key.

41. The computer as recited in claim 35, wherein the CPU is configured to form a generator seed from a CPU-specific secret and OS-specific data and to generate a private storage key based on a function of the generator seed.

42. A central processing unit comprising:

software identity register;

a boot log; and

processing means to process an atomic operation such that in an event that the atomic operation completes correctly, the software identity register is set to an identity of software code and the identity is appended to the boot log.

43. A computer system comprising:

a subscriber unit having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys, a manufacturer certificate supplied by a

manufacturer of the CPU, and a software identity register that holds an identity of the operating system, the subscriber unit being configured to submit a request over a network;

a content provider having storage to store content and a server to server the content to the subscriber, the content provider being configured to receive the request over the network, generate a challenge nonce, and return the challenge nonce to the subscriber unit; and

the subscriber unit being further configured to form an OS certificate containing the identity from the software identity register, information describing the operating system, the challenge nonce, and the CPU public key and to sign the OS certificate using the CPU private key, the subscriber unit returning the OS certificate and the CPU manufacturer certificate to the content provider for evaluation to determine whether to reject or fulfill the request.

44. The computer system as recited in claim 43, wherein the content provider is configured to determine whether to trust the identity in the OS certificate.

45. The computer system as recited in claim 43, wherein the content provider is configured to determine whether the challenge nonce returned in the OS certificate is the challenge nonce generated by the content provider.

46. The computer system as recited in claim 43, wherein the content provider is configured to verify the signature on the OS certificate using the CPU public key contained in the OS certificate.

47. The computer system as recited in claim 43, wherein the content provider is configured to determine whether the OS certificate and the manufacturer certificate contain an identical CPU public key.

48. The computer system as recited in claim 43, wherein the content provider is configured to verify a manufacturer signature on the manufacturer certificate.

49. The computer system as recited in claim 43, wherein the content provider is configured to determine whether to trust the manufacture of the CPU.

50. The computer system as recited in claim 43, wherein the content provider is configured to download the content specified in the request in an event that the content provider elects to fulfill the request.

51. The computer system as recited in claim 50, wherein the subscriber unit is configured to encrypt the content using a storage key derived in part from the identity of the operating system.

52. For execution on a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a software identity register, a computer program stored on one or more computer-readable storage media of the computer system, the program comprising:

executing an atomic operation to set an identity of the operating system into the software identity register of the CPU, wherein in an event that the atomic operation completes

correctly, the software identity register contains the identity of the operating system and in an event that the atomic operation fails to complete correctly, the software identity register contains a value other than the identity of the operating system; and

examining a content of the software identity register to verify the identity of the operating system.

53. For execution on a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys and a software identity register that holds an identity of the operating system, a computer program stored on one or more computer-readable storage media of the computer system, the program comprising:

forming an OS certificate containing the identity from the software identity register, information describing the operating system, and the CPU public key; and

signing the OS certificate using the CPU private key.

54. In a system having a subscriber unit and a content provider, in which the subscriber unit has a central processing unit (CPU) and an operating system (OS) and the CPU further includes a pair of private and public keys, a manufacturer certificate supplied by a manufacturer of the CPU, and a software identity register that holds an identity of the operating system, a computer program architecture stored on one or more computer-readable storage media resident at the subscriber unit and content provider for establishing a chain of trust between the subscriber unit and the content provider, the program comprising:

submitting a request from the subscriber unit to the content provider, the request specifying a particular content;

generating, at the content provider, a challenge nonce;

returning the challenge nonce from the content provider to the subscriber unit;

forming, at the subscriber unit, an OS certificate containing the identity from the software identity register, information describing the operating system, the challenge nonce, and the CPU public key and signing the OS certificate using the CPU private key;

passing the OS certificate and the CPU manufacturer certificate from the subscriber unit to the content provider; and

evaluating, at the content provider, the OS certificate and the CPU manufacturer at the content provider to determine whether to reject or fulfill the request.

55. For execution on a computer system having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys and a software identity register that holds an identity of the operating system, the computer system further maintaining a boot log that holds identities of software components that are currently executing, a computer program stored on one or more computer-readable storage media of the computer system, the program comprising:

forming a generator seed from a CPU-specific secret, a user-supplied seed, and OS-specific data from the boot log; and

generating a storage key based on a function of the generator seed.

56. A method for associating a level of trust with a user computer by a third party, the user computer having a central processing unit (CPU) and an operating system (OS), the CPU having a pair of private and public keys, a manufacturer certificate supplied by a manufacturer of the CPU, and a software identity register that holds an identity of an operating system executing on the CPU, the method comprising:

establishing a secure connection between the user computer and the third party;

generating, at the third party, a challenge nonce;

transmitting, by the third party, the challenge nonce to the user computer over the secure connection;

signing, by the user computer, an OS certificate and the challenge nonce using the CPU private key;

transmitting, by the user computer, the signed OS certificate and the signed challenge nonce to the third party over the secure connection; and

associating, by the third party, the level of trust for the user computer using the signed OS certificate.

57. The method as recited in claim 56, wherein the OS certificate comprises the software identity register.

58. The method as recited in claim 62, wherein the level of trust is based on the operating system identified by the software identity register.

59. The method as recited in claim 56, wherein the OS certificate comprises a boot log.

60. The method as recited in claim 56, wherein the OS certificate comprises a register containing a value associated with a boot log.

61. The method as recited in claim 56, wherein the OS certificate comprises identities of software components executing on the CPU.

62. The method as recited in claim 61, wherein the level of trust is based on the identities of the software components executing on the CPU.

63. The method as recited in claim 56, wherein the OS certificate comprises identities of device drivers executing on the CPU.

64. The method as recited in claim 63, wherein the level of trust is based on the identities of the device drivers executing on the CPU.

65. The method as recited in claim 56, further comprising:

submitting, by the user computer, a request to the third party for access to specific content;

evaluating, by the third party, whether to permit access based on the level of trust associated with the user computer.

66. The method as recited in claim 65, wherein the access comprises:

transmitting, from the third party, the specific content to the user computer through the secure connection.

67. The method as recited in claim 65, wherein the access comprises:

transmitting, from the third party, a storage key for the specific content to the user computer through the secure connection, wherein the specific content was previously stored on the user computer.

68. The method as recited in claim 67, wherein the specific content was obtained outside the secure connection.